Get into Programming with JavaScript

Axle Barr

Efficiency and economies of scale

Instead of drawing one line, why not draw four, five or eight

Instead of cooking one burger why not cook 5 or 10

Instead of measuring one seismic event, why not monitor several of them

HANDLING MULTIPLE ITEMS AT ONCE

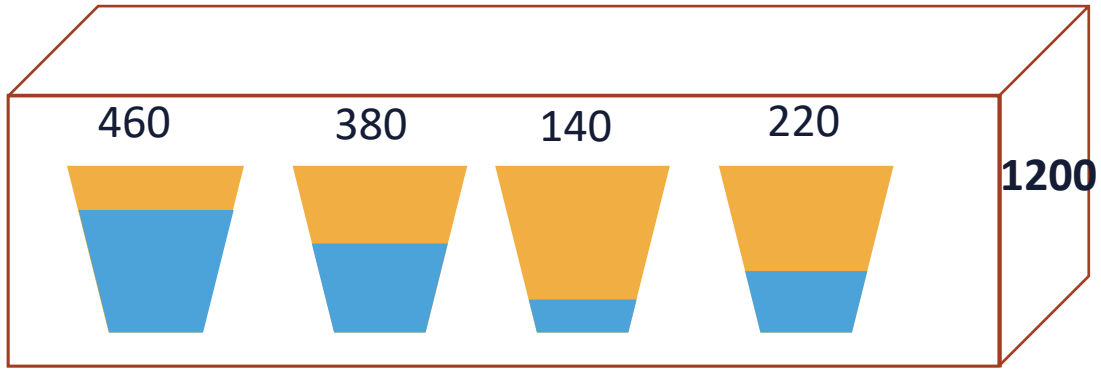
THE HUMBLE VARIABLE ON STEROIDS

Handling multiple values at once

- Variable store values
- As many as we want
- Different types, str, int, bool, float
- Limits
- An array is a collection of variables of the same type
- Array is very useful in programming
- Data structure

Put buckets in a barrel

We can pour the water from one or more buckets into a barrel or we can find a suitable container and put the buckets in there

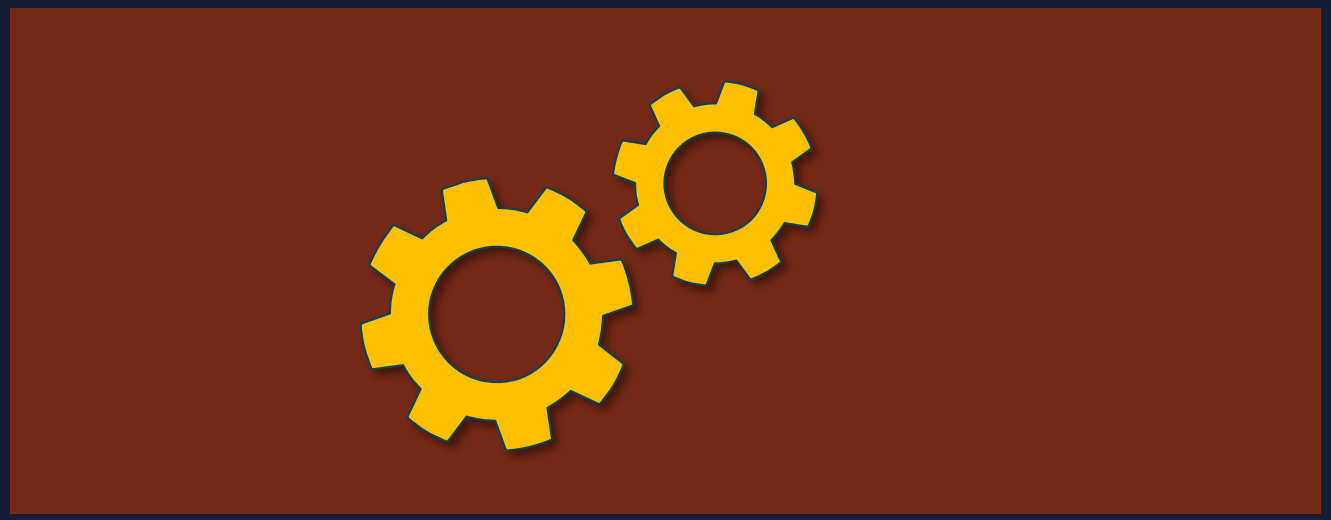


If we did the container option, then we can use the original buckets

CONTAINERIZED BUCKETS

EXAMPLES OF ARRAYS

Graphics and OST Goes Here:



WHICH MAILBOX BELONGS TO SAM

Or Sally or Jim or Axle

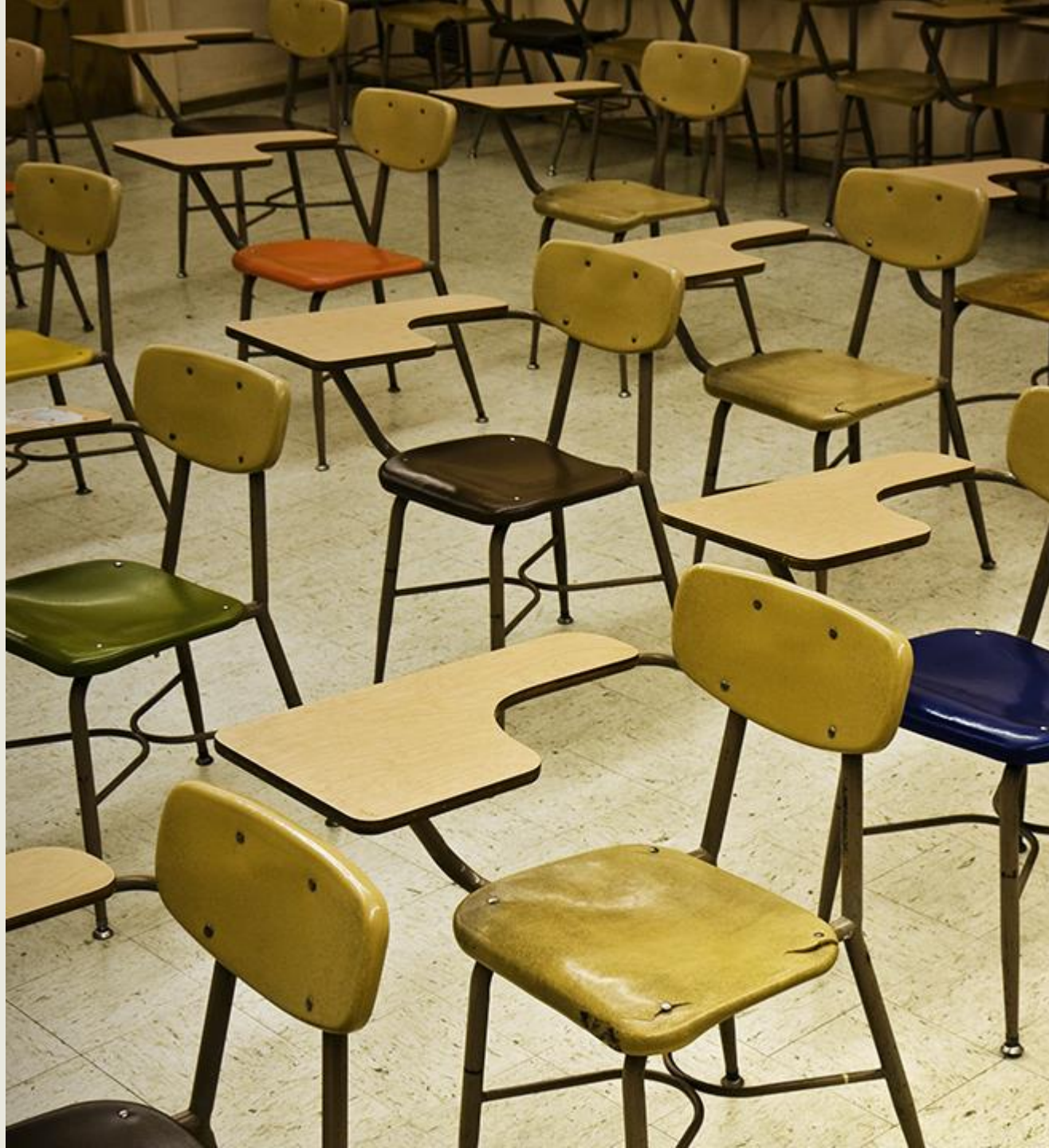
Mailboxes have numbers on them

WHERE DOES MARY SIT

Or Anthony or Clara or Axle

Each student can be identified by the row number and their seat in that row

We can say that Molly sits in the third row next the wall



WHERE DOES GEORGE LIVE

Or Collin or Chanelle or Axle

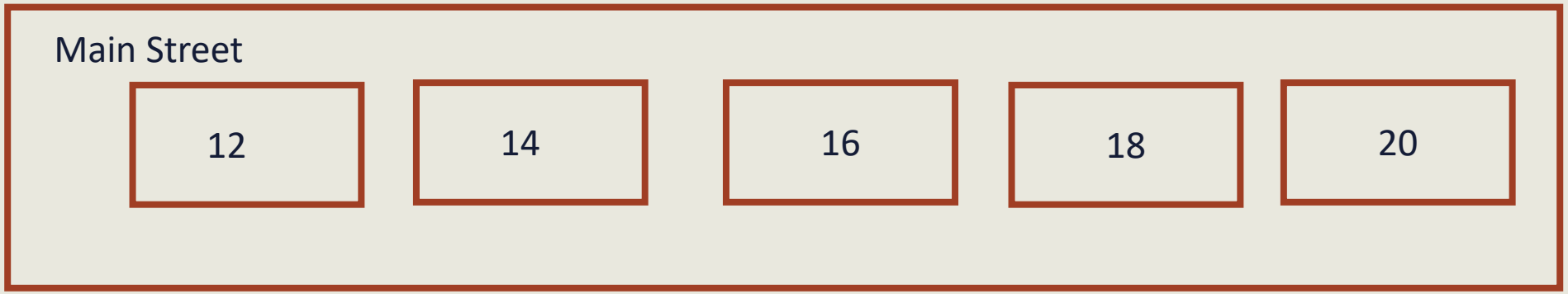
Houses on a street is more straight forward, we can say Lewis lives on 16 Main Street. Steve lives on 14 main street

IDENTIFY THE HOUSES

A typical street with houses on it

IDENTIFY THE HOUSES

A typical street with houses on it

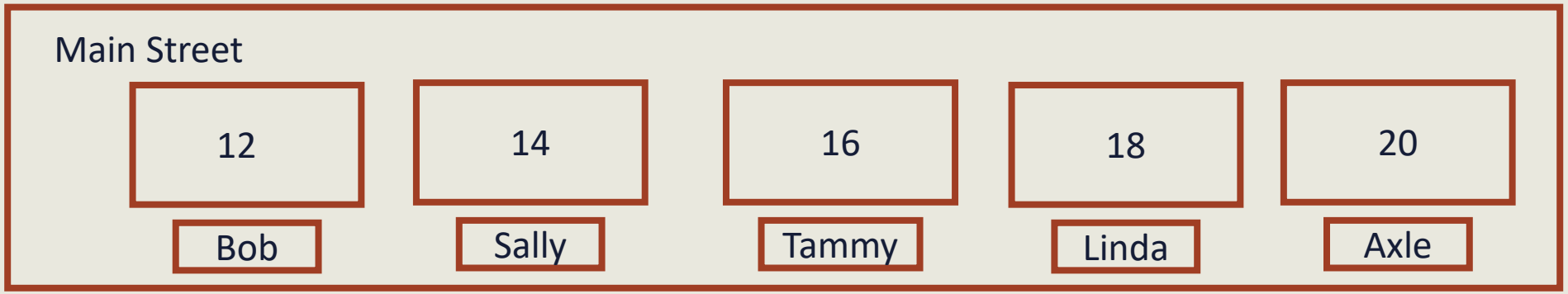


LARGE VARIABLES

- Think of an array as a very large variable, capable of holding not just one value, but as many as we want
- In order to be effective, this array or list must contain other variables of the same type, so all *ints*, or all *strings* etc.
- So if an array is a big box of other values, how do we identify individual values?
- The identification happens just like we can identify houses on a street, by their number or position, which is usually sequential

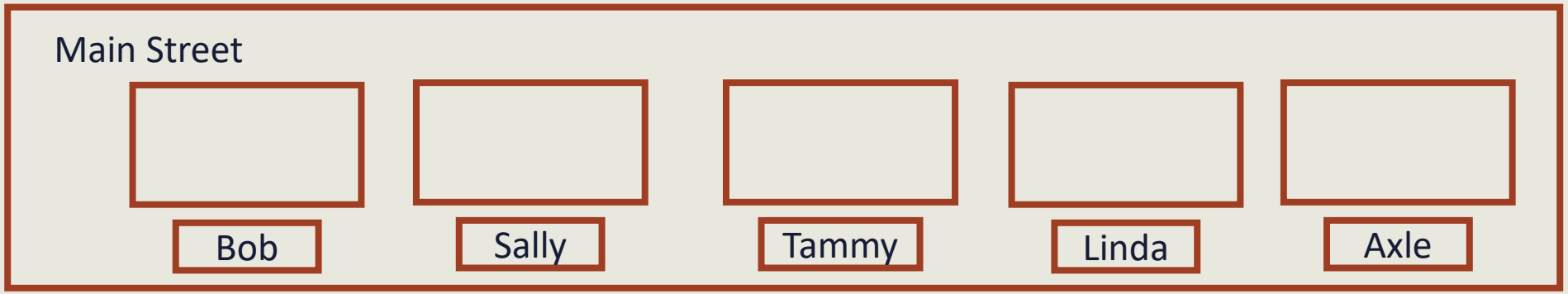
IDENTIFY THE HOUSES

A typical street with houses on it

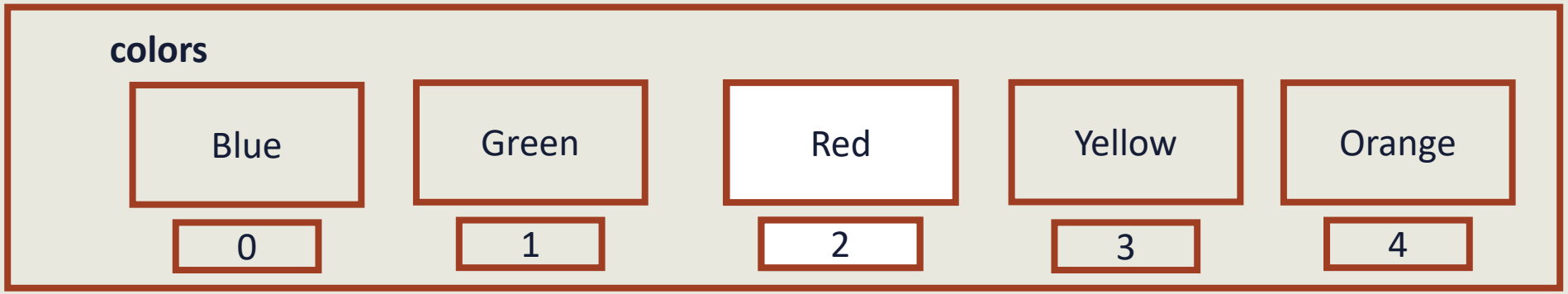


IDENTIFY THE HOUSES

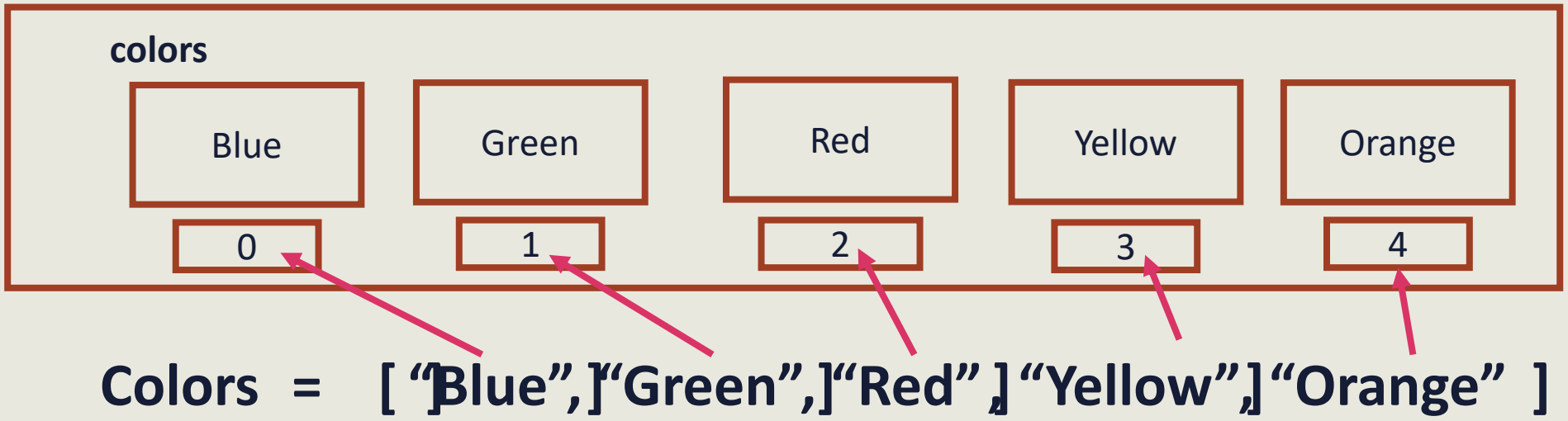
A typical street with houses on it



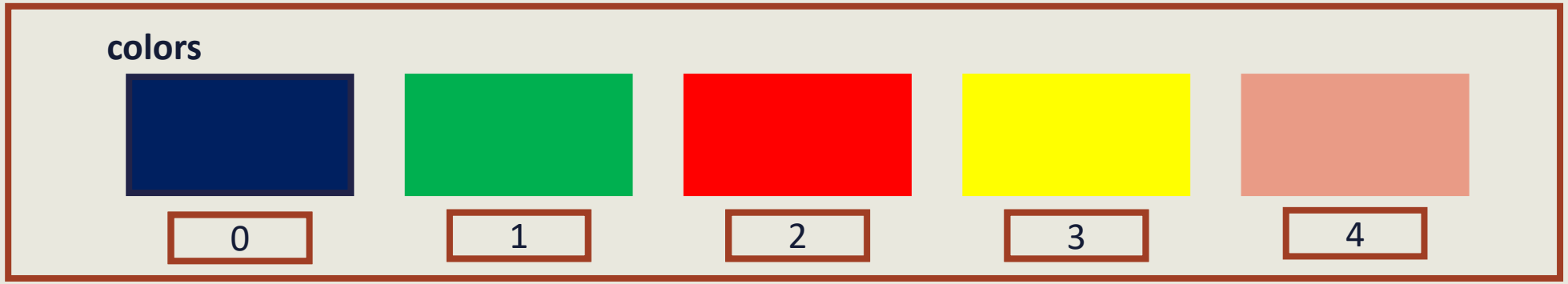
AN ARRAY OF COLOURS



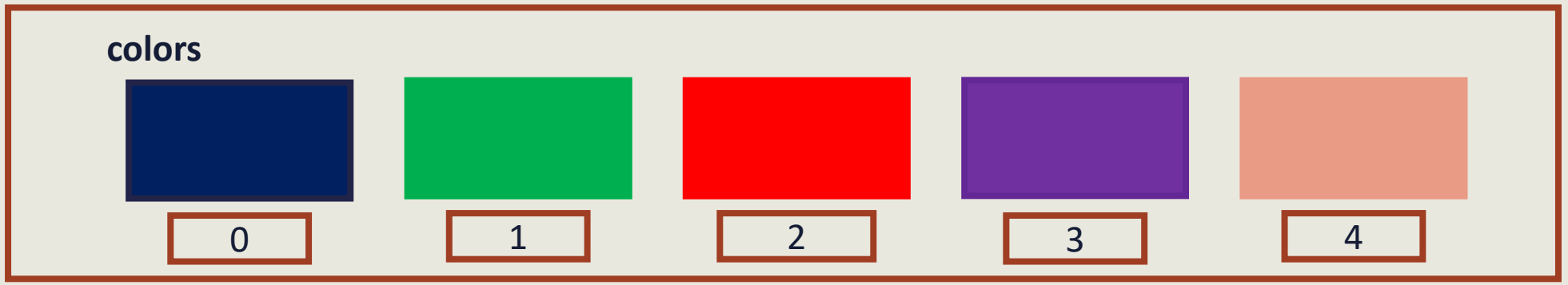
AN ARRAY OF COLOURS



AN ARRAY OF COLOURS

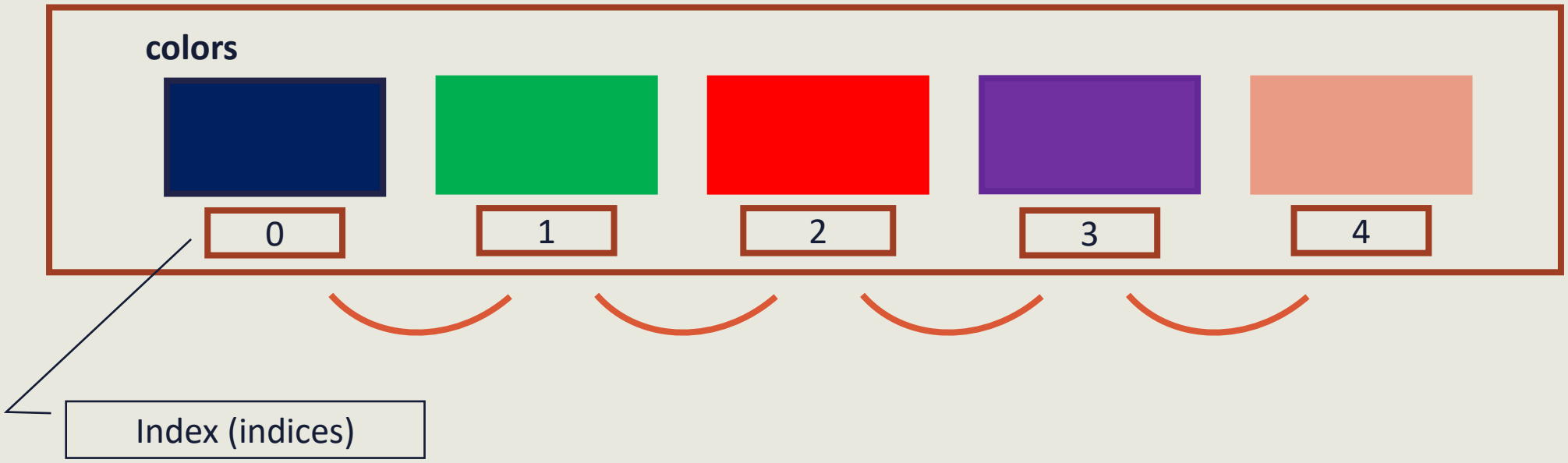


AN ARRAY OF COLOURS



The array with the name of ***colors*** contains 5 elements numbered from 0 to 4

AN ARRAY OF COLOURS



DECLARING AND USING ARRAYS

01

```
const colors = ["blue", "green", "yellow", "red", "orange"];
printOut = colors[2];
```

02

```
const colors = ["blue", "green", "yellow", "red", "orange"];
printOut = colors[4];
```

03

```
const colors = ["blue", "green", "yellow", "red", "orange"];
colors.push("purple");
printOut = colors[5];)
```

ITERATING THROUGH THE ARRAY

01

```
const colors = ["blue", "green", "yellow", "red", "orange"];
colors.push("purple");
printOut = colors[5];)
```

02

```
const colors = ["blue", "green", "yellow", "red", "orange"];
colors.push("purple");
for (let i=0; i <= 5; i++) {
  printOut += colors[i] + "<br>";
}
```

03

```
for (let i=0; i < colors.length; i++) {
  printOut += colors[i] + "<br>";
}
```

04

```
for (let i=0; i < colors.length; i++) {
  printOut += colors[i] + " is my favorite!<br>";
}
```

ITERATING THROUGH THE ARRAY

01

```
const colors = ["blue", "green", "yellow", "red", "orange"];
for (let x in colors) {
  printOut += colors[x];
}
```

02

```
for (let x in colors) {
  printOut += colors[x] + ", ";
}
```

03

```
for (let x in colors) {
  printOut += colors[x] + "<br />";
}
```

04

```
for (let color in colors) {
  printOut += colors[color] + "<br />";
}
```

ITERATING THROUGH THE ARRAY

01

```
let total = 0;
const cart = [ 100, 200, 400, 300 ];
for (let item in cart) {
  total+=cart[item];
}
printOut = "Total in cart " + total;
```

02

```
for (let item in cart) {
  total+=cart[item];
  printOut += "Item " + item + "<br />";
}
printOut += "Total in cart " + total;
```

03

```
for (let item in cart) {
  total+=cart[item];
  printOut += "Item " + item + " was " + cart[item] + "<br />";
}
printOut += "Total in cart " + total;
```

ARRAY FACTS

Know the following about arrays:

- Name
- Type of data stored
- Length (size)
- Identify elements
- Insert elements
- Remove elements
- *Other methods and functions*

MORE ARRAY FACTS

- Delete from an array means using a function
- You can delete any element, but you must identify which one
- Insert and delete work similarly
- Once you insert or delete the length of the array changes
- It is possible to insert/delete multiple elements at once
- An error will occur if you try to access an element that does not exist

More ARRAY Operations

01

```
const colors = ["blue", "green", "yellow", "red", "orange"];
colors.unshift("white");
for (let color in colors) {
  printOut += colors[color] + "<br />";
}
```

02

```
const colors = ["blue", "green", "yellow", "red", "orange"];
colors.splice(2,0,"white");//0 means do not delete anything
for (let color in colors) {
  printOut += colors[color] + "<br />";
}
```

03

```
const colors = [];
colors[0] = "blue";
colors[1] = "red";
colors[2] = "green";
```

More ARRAY Operations

01

```
const colors = ["blue", "green", "yellow", "red", "orange"];
printOut = colors.join(" ");
```

02

```
const colors = ["blue", "green", "yellow", "red", "orange"];
printOut = colors.join(", ");
```

03

```
const colors1 = ["red", "orange", "yellow", "green"];
const colors2 = ["blue", "indego", "violet"];
rainbow = colors1.concat(colors2);
printOut = rainbow.join(", ");
```

A SIMPLE ARRAY

Use arrays instead of many variables

color1 = “red”

color2 = “green”

color3 = “blue”

```
colors = [" red ", "green", "blue"]
```

ACCESSING INDIVIDUAL ELEMENTS

For each element in the array I want to print its value, so red , then green and finally blue

```
colors = [" red ", "green", "blue"]
```

```
for (let color in colors) {
  printOut += colors[color] + "<br />";
}
```

```
for (let color in colors) {
    if(colors[color] == “blue”)
        printOut = “found the blue color”;
}
```

FINDING INDIVIDUAL ELEMENTS

```
const colors = ["red", "orange", "yellow", "green", "blue", "indego", "violet"];
let colorToFind = prompt("Which color are you looking for? ");
for (let i=0; i < colors.length; i++) {
        if(colorToFind == colors[i]){
                printOut = "We found your color";
        }
        else{
                printOut = "We could not find your color";
        }
}
```

colors = [" red ", "green", "blue"]

FINDING INDIVIDUAL ELEMENTS

```
const colors = ["red", "orange", "yellow", "green", "blue", "indego", "violet"];
let colorToFind = prompt("Which color are you looking for? ");
for (let i=0; i < colors.length; i++) {
        if(colorToFind == colors[i]){
                printOut = "We found your color";
                break;
        }
        else{
                printOut = "We could not find your color";
        }
}
```

colors = [" red ", "green", "blue"]

FINDING INDIVIDUAL ELEMENTS

```
const colors = ["red", "orange", "yellow", "green", "blue",  "violet"];
let colorToFind = prompt("Which color are you looking for? ");
for (let i=0; i <= colors.length; i++) {
        if(colorToFind == colors[i]){
                printOut = "We found your color";
                break;
        }
}
```

colors = [" red ", "green", "blue"]

FINDING INDIVIDUAL ELEMENTS

```
const colors = ["red", "orange", "yellow", "green", "blue",  "violet"];
let colorToFind = prompt("Which color are you looking for? ");
for (let i=0; i <= colors.length; i++) {
        if(colorToFind == colors[i]){
                printOut = "We found your color";
                break;
        }
}
if (printOut == "")
        printOut += "We could not find your color";
```

colors = [" red ", "green", "blue"]

A SIMPLE ARRAY

Sorting Arrays

```
const colors =
    ["red", "orange", "yellow", "green", "blue", "indego", "violet"];
colors.sort();
printOut = colors.join(", ");
```

A SIMPLE ARRAY

Sorting Arrays

```
const colors =
    ["red", "orange", "yellow", "green", "blue", "indego", "violet"];
colors.reverse();
printOut = colors.join(", ");
```

Sorting Arrays

A SIMPLE ARRAY

```
const colors =
    ["red", "orange", "yellow", "green", "blue", "indego", "violet"];
const newColors = colors.slice(4);
printOut = newColors.join(", ");
```

Sorting Arrays

```
const colors =
    ["red", "orange", "yellow", "green", "blue", "indego", "violet"];
printOut = colors.toString();
```

A SIMPLE ARRAY

A Special Note on Arrays

```
const colors = ["red", "orange", "yellow“];
colors.pop()
printOut = colors.toString();
```

Should not be allowed

A Special Note on Arrays

```
const colors = ["red", "orange", "yellow", "green"];
colors.sort();
printOut = colors.join(", ");
```

What if we wanted to see the *before* and *after*

A Special Note on Arrays

```
const colors = ["red", "orange", "yellow", "green"];
printOut = "Before: " + colors.join(", ");
colors.sort();
printOut += "<br />After: " + colors.join(", ");
```